MULTI-LINGUAL DEEP NEURAL NETWORKS FOR LANGUAGE RECOGNITION

Luis Murphy Marcos

University of Puerto Rico
Mayagüez Campus
Mayagüez, Puerto Rico

Frederick Richardson

MIT Lincoln Laboratory
Human Language Technology Group
Lexington, MA USA

ABSTRACT

Multi-lingual feature extraction using bottleneck layers in deep neural networks (BN-DNNs) has been proven to be an effective technique for low resource speech recognition and more recently for language recognition. In this work we investigate the impact on language recognition performance of the multi-lingual BN-DNN architecture and training configurations for the NIST 2011 and 2015 language recognition evaluations (LRE11 and LRE15). The best performing multi-lingual BN-DNN configuration yields relative performance gains of 50% on LRE11 and 40% on LRE15 compared to a standard MFCC/SDC baseline system and 17% on LRE11 and 7% on LRE15 relative to a single language BN-DNN system. Detailed performance analysis using data from all 24 Babel languages, Fisher Spanish and Switchboard English shows the impact of language selection and the amount of training data on overall BN-DNN performance.

Index Terms: language recognition, multi-lingual deep neural network, multi-lingual bottleneck features

1. INTRODUCTION

Multi-lingual modeling has gained renewed interest particularly for low resource automatic speech recognition (ASR). In the recent IARPA Babel OpenKWS15 Evaluation [1], top performing systems used language independent feature representations extracted from a multi-lingual deep neural networks with bottleneck layers (BN-DNNs) to train language dependent DNNs for the surprise language [2]. Multi-lingual BN-DNN features have also been shown to perform well for language recognition [3] and proved to be a very effective approach in the NIST 2015 language recognition evaluation (LRE15) open training condition [4, 5].

In this work we evaluate the impact of different architectures and training configurations on the performance of the multi-lingual BN-DNN approach on LRE11 and the more recent LRE15. Data from the 24 Babel build packs [1], LDC Switchboard English [6] and LDC Fisher Spanish [7] are used to train multi-lingual BN-DNNs. The results reported here highlight the importance of selecting appropriate languages and sufficient amounts of data per a language in order to obtain good language recognition performance.

THIS WORK WAS SPONSORED BY THE DEPARTMENT OF DEFENSE UNDER AIR FORCE CONTRACT FA8721-05-C-0002. OPINIONS, INTERPRETATIONS, CONCLUSIONS, AND RECOMMENDATIONS ARE THOSE OF THE AUTHORS AND ARE NOT NECESSARILY ENDORSED BY THE UNITED STATES GOVERNMENT.

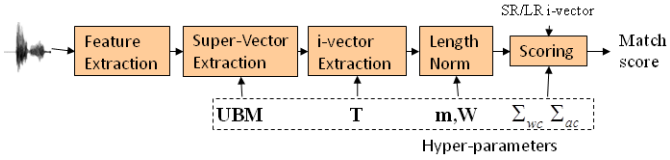

Fig. 1. I-vector system architecture

2. I-VECTOR SYSTEM

Most state-of-the-art language recognition systems are based on the i-vector framework [8] depicted in Figure 1. The i-vector system uses a Gaussian mixture model (GMM) which is often referred to as the universal background model (UBM) to extract zero'th and first order statistics from the input sequence of feature vectors. A super vector created by stacking the first order statics is transformed to a lower dimensional sub-space using a linear transformation that depends on the zeroth order statistics (see [9] for more details). This transformation requires a total variability matrix $\mathbf{T}$ which is estimated from a large set of super-vectors using an EM-algorithm [9].

The i-vector is treated as a single low dimensional representation of a waveform that contains speaker, language, session and channel information. With a sufficient number of recorded sessions across all languages it is possible to estimate a full rank within class covariance matrix ($\mathbf{\Sigma}_{\text{wc}}$). It is more difficult to estimate a full rank across class covariance matrix ($\mathbf{\Sigma}_{\text{ac}}$) because of the relatively small number of languages available compared to the i-vector dimension. In this work we use PPCA to estimate a full rank $\mathbf{\Sigma}_{\text{ac}}$ matrix. Another important set of parameters for an i-vector system are the mean vector $\mathbf{m}$ and whitening matrix $\mathbf{W}$ which are used to transform the i-vectors to a unit normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ over a pool of data before applying length normalization [10]. Typically $\mathbf{m}$ and $\mathbf{W}$ are estimated on the same data used to estimate $\mathbf{\Sigma}_{\text{wc}}$ and $\mathbf{\Sigma}_{\text{ac}}$. I-vector whitening and lenght normalization are generally applied before i-vector scoring.

An effective technique for computing the likelihood ratio that an i-vector $\mathbf{z}_i$ represents speech data from the same language as an i-vector language model $\mathbf{z}_l$ ($\mathcal{H}_s$) or from some other language ($\mathcal{H}_d$) is probabilistic linear discriminant analysis (PLDA). The PLDA likelihood ratio is given by

$$\frac{p(\mathbf{z}_i, \mathbf{z}_l|\mathcal{H}_s)}{p(\mathbf{z}_i, \mathbf{z}_l|\mathcal{H}_d)}$$

which can be computed using the "2 covariance model" described in [11] with the hyper parameters $\mathbf{\Sigma}_{\text{wc}}$ and $\mathbf{\Sigma}_{\text{ac}}$.

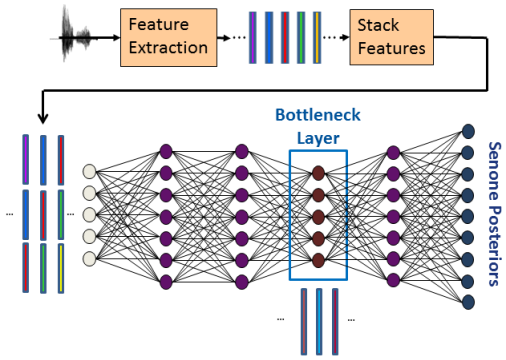

Fig. 2. BN-DNN architecture for speaker or language recognition

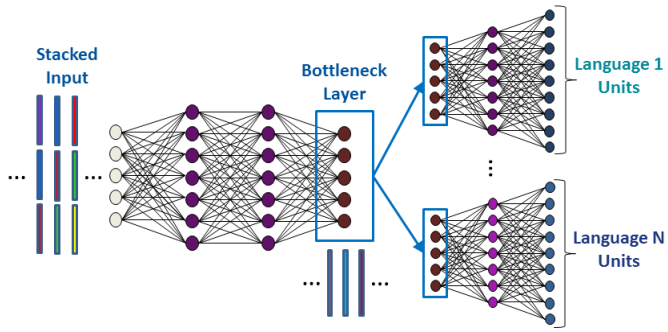

Fig. 3. Multi-lingual BNF architecture

3. BOTTLENECK DNN

The feed forward DNN depicted in Figure 2 has a narrow layer - the bottleneck (BN) layer - that is used for extracting features that are then used to train another classifier [12]. In this work, a 7 layer BN-DNN is used where the input is an 819 dimensional stacked feature vector and the output is a vector of senone posterior probability estimates. The number of senones depends on the language corpora and the ASR system used to create senone alignments for the data. The input vector consists of 21 stacked 39 dimensional feature vectors where each feature vector consists of 13 perceptual linear predictor coefficients (PLPs) normalized using utterance level feature warping along with their first and second order derivatives [13, 14]. In this work we use a linear BN layer without an offset which is equivalent to replacing the layer before and after the BN with a single weight matrix with a rank no greater than the BN dimension [15]. Utterance level feature warping is also applied to the BN-DNN features.

4. MULTILINGUAL DNN

The multi-lingual DNN architecture used in this work is depicted in Figure 3. The first 5 hidden layers of the DNN are shared across all languages in the training set and the last two layers are unique for each language. A modified version of the standard stochastic gradient descent training algorithm draws a 512 sample mini-batch from each language in sequence. Currently all languages are sampled from equally which results in the least represented language determining the total amount of training data used in each epoch.

The DNN multi-lingual architecture shown in Figure 3 is very similar to BN-DNN architecture described in Section 3 except that the last two output layers are different for each language. A straight forward implementation of the modified SGD training algorithm with Theano [16] uses a common set of Theano shared variables for the language independent layers and unique Theano shared variables for the language dependent output layers. The gradient of the language dependent cost functions are then easily computed by applying the Theano "grad" function to the cross entropy between the ground truth labels and the output softmax for each language. A separate Theano function is then created for processing mini-batches in each language with updates to the corresponding set of language dependent and language independent Theano gradient parameters.

5. EXPERIMENTAL SETUP

The multi-lingual bottleneck DNNs used in this work were trained with different combinations of the 24 IARPA Babel languages [1] as well as LDC Fisher Spanish and Switchboard English [7, 6]. The 26 language corpora are summarized in Table 1. A Kaldi [17] "tri4a" system was trained for each language to obtain frame level labels (senones or triphone state clusters) using the pronunciation lexicon provided for the corresponding corpora. The number of seneones for each language is also reported in Table 1. The DNN training configurations including the languages and amount of data per a language are summarized in Table 2.

Two language recognition tasks are used for evaluating the multi-lingual bottleneck systems. The first is the NIST 2011 Language Recognition Evaluation (LRE11) [18] which includes 24 languages from both telephone and broadcast sources and has test durations of 3, 10, and 30 seconds. The second is the more recent NIST 2015 Language Recognition Evaluation (LRE15) [19] which consists of 20 languages partitioned in 6 language clusters: Arabic, Chinese, English, French, Iberian and Slavic. The LRE11 and LRE15 performance scoring metric is the NIST C_{avg} cost function. LRE11 is scored separately for each duration (3, 10 and 30 seconds) and LRE15 is scored using the average of the C_{avg} for each language cluster. Details on the LRE2011 and LRE15 training and development data can be found in [20] and [4] respectively.

Following the discussion in [4], LRE15 performance is given both with and without the French language cluster and the language models are trained only on the fixed LRE15 training data. LRE15 performance is computed excluding any cuts that are from the 24 Babel corpora listed in Table 1.

6. EXPERIMENTS

Except where explicitly stated otherwise, the BN-DNN configurations in the following experiments use 7 hidden layers where the second to last hidden layer is an 80 dimensional linear bottleneck and all other hidden layers have 1024 nodes with sigmoid activation functions. In all cases the i-vector system uses a 2048 component Gaussian mixture model (GMM) and a 600 dimensional total variability sub-space. The i-vectors are scored using PLDA described in Section 2 and a discriminative Gaussian backend trained on development data is used for score calibration (see [20] and [4] for more details). All systems use speech activity segmentation generated using a GMM based speech activity detector. Finally, the front-end feature extraction for the baseline system uses 7 static cepstra appended to 49 shifted delta cepstra (SDC) for a total of 56 features.

Language	LDC or Babel corpora	Senones
English	Switchboard 1 release 2	4144
Spanish	Fisher Spanish	3956
Cantonoese	**IARPA-babel101b-v0.4c**	**4615**
Assamese	IARPA-babel102b-v0.5a	4677
Bengali	IARPA-babel103b-v0.4b	4773
Pashto	**IARPA-babel104b-v0.bY**	**4824**
Turkish	**IARPA-babel105b-v0.4**	**4771**
Tagalog	**IARPA-babel106b-v0.2g**	**4700**
Vietnamese	**IARPA-babel107b-v0.7**	**4664**
Haitian	IARPA-babel201b-v0.2b	4822
Swahili	IARPA-babel202b-v1.0d	4616
Lao	IARPA-babel203b-v3.1a	4733
Tamil	IARPA-babel204b-v1.1b	4341
Kurmanji	IARPA-babel205b-v1.0a	4545
Zulu	IARPA-babel206b-v0.1e	4519
Tok Pisin	IARPA-babel207b-v1.0e	4699
Cebuano	IARPA-babel301b-v2.0b	4663
Kazakh	IARPA-babel302b-v1.0a	4559
Telugu	IARPA-babel303b-v1.0a	4412
Lithuanian	IARPA-babel304b-v1.0b	4821
Guarani	IARPA-babel305b-v1.0c	4513
Igbo	IARPA-babel306b-v2.0c	4523
Amharic	IARPA-babel307b-v1.0b	4713
Mongolian	IARPA-babel401b-v2.0b	4543
Javanese	IARPA-babel402b-v1.0b	4669
Dholuo	IARPA-babel403b-v1.0b	4644

Table 1. Language corpora and number of senones used in experiments (Babel "BP" languages are in bold)

Note that "FER" reported in the tables is the lowest BN-DNN frame error rate on held out validation data for the last training epoch.

6.1. BN-DNN Configurations

The following analysis evaluates the performance of i-vector systems trained with BN features extracted from uni-lingual or multi-lingual BN-DNNs with different architectures, different amount of training data and different sets of corpora.

Table 3 shows the impact of the amount of training data and the BN layer dimension on performance for a BN-DNN trained only on Switchboard English. Increasing the amount of trainnig data from 100 to 300 hours improves the average relative performance by 5% for LRE11 and 8% for LRE15 for the 64 dimensional BN and 11% for LRE11 and 5% for LRE15 for the 80 dimensional BN. Increasing the BN dimension from 64 to 80 on the other hand degrades performance by 5% for LRE11 and improves performance by 2% for LRE15 for 100 hours of training data and improves performance by 2% for LRE11 while having no impact on performance for LRE15 for 300 hours of training data. This suggests that the amount of data has a bigger impact on performance than the width of the BN layer and a wider BN layer can degrade performance unless there is a sufficient amount of training data. All remaining experiments reported here use an 80 dimensional linear BN layer.

Table 4 shows the impact on performance of the hidden layer dimension for a BN-DNN trained with the 7L-420hr configuration. On average there is a relative improvement of 16% for LRE11 and 6% for LRE15 increasing the hidden layer dimension from 512 to 1024 nodes and a relative improvement of 7% on LRE11 and 3% on LRE15 increasing the hidden layer dimension from 1024 to 2048 nodes. The small relative gain for the 2048 node hidden layer BN-DNN and the increased size and computation requirements may not be a reasonable trade off for some applications. All remaining experiments reported here use 1024 dimensional sigmoidal hidden layers along with the 80 dimensional linear BN layer.

Table 5 gives the performance for training a BN-DNN with the 24 Babel languages on different amounts of data per a language. Increasing the amount of data from 10 to 20 hours per a language gives an average relative performance gain of 9% and 4% on LRE11 and LRE15 respectively. Increasing the amount of data further from 20 to 40 hours per a language yields an additional 5% and 3% average relative gain on each task. Clearly there are diminishing returns as one increases the amount of data per a language which also has a significant impact on the amount of time required to train a BN-DNN.

6.2. BN-DNN Results

Table 6 summarizes the performance for the various BN-DNN ivector systems described in Section 6.1 as well as the baseline SDC i-vector system. As shown in our prior work [13, 14], all the BN-DNN i-vector systems give substantial gains in performance relative to the SDC i-vector baseline. In this new work, the largest gains are coming from the 7L-420hr BN-DNN which gives an average relative gains of 50% on LRE11 and 40% on LRE15. Compared to the single language 1L-300h BN-DNN the relative gains for the 7L-420hr BN-DNN are 17% on LRE11 and 7% on LRE15.

It is interesting that we are not seeing any gain from the 24L-960hr system which uses both more languages and more data than the 7L-420hr system. One possible explanation at least for LRE15 is that the 24 Babel languages do not cover many of the 6 language clusters including English and Iberian which are at least partially covered by the 7L-420hr training configuration which includes Switchboard English and Fisher Spanish data. In fact the only other language cluster covered by the 26 Babel language corpora is French from the Babel Haitian data.

Table 7 gives the LRE15 breakdown per a language cluster for the 7L-420hr and 24L-960hr systems. The largest degradations in performance (from 8% to 18%) come from the English, Slavic and Iberian language clusters likely due to the lack of Switchboard English and Fisher Spanish data in the 24L-960hr training configuration. There is a small relative improvement of about 4% for the French language cluster which may be due to the additional Babel Haitian data in the 24L-960hr training configuration.

The 26L-520hr training configuration includes the English and Spanish data missing from the 24L-960hr configuration, but the amount of data per a language is reduced from 40 hours to 20 hours which may explain the lackluster performance. Unfortunately we do not currently have results for a 26 language configuration with 40 hours per a language (or 1040 hours of data total) which presumably would yield better performance. Currently this larger configuration would take considerably longer to train (the 24L-960hr DNN in Table 6 required almost two weeks). In the future we hope to parallelize and optimize our SGD implementation.

7. CONCLUSIONS

In this work we have shown that multi-lingual BN-DNN features can give significant performance gains even when compared to single language BN-DNN features. The 7L-420hr training configuration yields a 17% relative improvement on LRE11 when compared to the single language BN-DNN and a 50% relative improvement

Training	Corpora	#Lang	Hours / lang	Total hours
1L-100hr	SWB English	1	100	100
1L-300hr	SWB English	1	300	300
5L-300hr	Babel BP Languages	5	60	300
7L-420hr	Babel BP, SWB Eng, FSH Span	7	60	420
24L-240hr	All Babel languages	24	10	240
24L-480hr	All Babel languages	24	20	480
24L-960hr	All Babel languages	24	40	960
26L-520hr	All Babel, SWB Eng, FSH Span	26	20	300

Table 2. DNN training configurations used for experiments

Training	BN	FER	LRE11			LRE15	
			30 Sec	10 Sec	3 Sec	Avg	w/o French
1L-100hr	64	63.8	2.24	6.33	16.7	18.7	12.8
1L-100hr	80	54.3	2.44	6.63	16.9	18.4	12.5
1L-300hr	64	50.4	2.06	6.15	16.2	17.5	11.7
1L-300hr	80	50.5	2.05	5.92	16.0	17.5	11.8

Table 3. LRE11 and LRE15 C_{avg} performance for 64 and 80 dimension BN-DNN trained with 100 and 300 hours of Switchboard data.

Training	Hidden layer	FER	LRE11			LRE15	
			30 Sec	10 Sec	3 Sec	Avg	w/o French
7L-420hr	512	60.9	2.00	6.01	15.5	17.3	11.7
7L-420hr	1024	60.7	1.58	4.91	14.1	16.6	10.7
7L-420hr	2048	58.6	1.38	4.62	13.7	16.3	10.3

Table 4. LRE11 and LRE15 C_{avg} using 512, 1024 and 2048 node hidden layers with the 7L-420hr training configuration.

Training	Hours / lang	Total hours	FER	LRE11			LRE15	
				30 Sec	10 Sec	3 Sec	Avg	w/o French
24L-240hr	10	240	60.6	2.20	5.94	16.1	18.2	12.4
24L-480hr	20	480	58.4	1.93	5.41	15.1	17.5	11.8
24L-960hr	40	960	57.1	1.78	5.11	15.0	17.0	11.4

Table 5. LRE11 and LRE15 C_{avg} using 10, 20 and 40 hours and 24 Babel languages.

Training	FER	LRE11			LRE15	
		30 Sec	10 Sec	3 Sec	Avg	w/o French
Baseline	N/A	4.34	10.0	21.4	25.5	19.7
1L-300hr	50.5	2.05	5.92	16.0	17.5	11.8
5L-300hr	60.4	1.76	5.23	‘14.8	17.0	11.0
7L-420hr	60.7	**1.58**	**4.91**	**14.1**	**16.6**	**10.7**
24L-960hr	57.1	1.78	5.11	15.0	17.0	11.4
26L-520hr	58.5	1.69	5.12	14.6	17.3	11.4

Table 6. LRE11 and LRE15 C_{avg} for 80 dimensional BNF and different training configuration

Cluster	7L-420hr	24L-960hr
Arabic	17.8	18.2
Chinese	7.82	7.95
English	8.58	10.1
French	46.2	44.5
Iberian	16.8	18.2
Slavic	2.54	2.80
Average	16.6	17.0

Table 7. LRE15 performance comparison for 7L-420hr and 24L-960hr training configurations

when compared to the MFCC/SDC baseline. The relative gains are substantially smaller on LRE15 and there is some indication that this has to do with the language mismatch between the languages used to train the BN-DNN and the LRE15 language clusters. Including English and Spanish in the BN-DNN training gives a significant improvement in performance compared to using large amounts of only the 24 Babel languages. This indicates that it may be possible to achieve more gains on the Arabic and Chinese cluster by adding additional ASR corpora such as Callhome Egyptian Arabic or HKUST Mandarin Chinese.

One topic not addressed in this work is the inherent limitations of the fixed training condition for LRE15 which may not represent all dialects of each language cluster at test time equally well. Through multi-condition BN-DNN training and various data augmentation strategies it is possible compensate for some mismatches between training and test data channel conditions [21, 22], but it is not clear how one would apply the same technique to artificially add phonotactic variability to speech for example to account for formal and informal versions of language dialect (see [4] for a discussion on the LRE15 French cluster).

8. REFERENCES

[1] Mary P. Harper, "https://www.iarpa.gov/index.php/research-programs/babel," .

[2] J. Cui, B. Kingsbury, B. Ramabhadran, A. Sethy, K. Audhkhasi, X. Cui, E. Kislal, L. Mangu, M. Nussbaum-Thom, M. Picheny, Z. Tuske, P. Golik, R. Schluter, H. Ney, M. Gales, K. Knill, A. Ragni, H. Wang, and P. Woodland, "Multilingual representations for low resource speech recognition and keyword search," in *Proc. of IEEE ASRU*, 2015.

[3] K. Vesely, M. Karafiat, F. Grez, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Proc. of IEEE SLT Workshop*, 2012.

[4] P. Torres-Carrasquillo, N. Dehak, E. Godoy, D. Reynolds, F. Richardson, S. Shum, E. Singer, and D. Sturim, "The MITLL NIST LRE 2015 language recognition system," in *Proc. of IEEE Odyssey*, 2016.

[5] O. Plchot, P. Matejka, R. Fer, O. Glembek, O. Novotny, J. Pesan, K. Vesely, L. Ondel, M. Karafiat, F. Grezl, S. Kesiraju, L. Burget, N. Brummer, A. Swart, S. Cumani, S.Mallidi, and R. Li, "BAT system description for NIST LRE 2015," in *Proc. of IEEE Odyssey*, 2016.

[6] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc. of ICASSP*, 1992, pp. 517–520.

[7] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: a resource for the next generations of speech-to-text," in *Proc. of LREC*, 2004.

[8] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via ivectors and dimensionality reduction," in *Proc. of Interspeech*, 2011, pp. 857–860.

[9] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, "Front end factor analysis for speaker verification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 19, no. 4, pp. 788–798, may 2011.

[10] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. of Interspeech*, 2011, pp. 249–252.

[11] N. Brummer and E. de Villiers, "The speaker partitioning problem," in *Proc. of IEEE Odyssey*, 2010.

[12] V. Fontaine, C. Ris, and J.-M. Boite, "Nonlinear discriminant analysis for improved speech recognition," in *Eurospeech*, 1997.

[13] F. Richardson, D. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," in *Proc. of Interspeech*, 2015.

[14] F. Richardson, Douglas Reynolds, and Najim Dehak, "Deep neural network approaches to speaker and language recognition," in *IEEE Signal Processing Letters*, 2015.

[15] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization," in *Proc. of ICASSP*, 2014, pp. 185–189.

[16] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, 2016.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J Silovsky, G. Stemmer, and K. Vesel, "The kaldi speech recognition toolkit," in *Proc. of IEEE ASRU*, 2011.

[18] "The 2011 NIST language recognition evaluation plan," 2011.

[19] "The 2015 NIST language recognition evaluation plan," 2015.

[20] E. Singer, P. Torres-Carrasquillo, D. Reynolds, A. McCree, F. Richardson, N. Dehak, and D. Sturim, "The MITLL NIST LRE 2011 language recognition system," in *Proc. of IEEE Odyssey*, 2011, pp. 209–215.

[21] Alan McCree, Gregory Sell, and Daniel Garcia-Romero, "Augmented data training of joint acoustic/phonotactic DNN i-vectors for NIST LRE15," in *Proc. of IEEE Odyssey*, 2016.

[22] F. Richardson, B. Nemsick, and D. Reynolds, "Channel compensation for speaker recognition using MAP adapted PLDA and denoising DNNs," *to appear in Odyssey*, 2016.